Primera Práctica

Teoría de Autómatas y Lenguajes Formales I

Escuela Politécnica Superior, Universidad Autónoma de Madrid

ENTREGA: la semana del 4 de abril, en las horas de prácticas

Curso 2004-2005

1 Introducción

Este año habrá una práctica incremental, que conducirá, al final del cuatrimestre, al desarrollo de un sistema de visualización de imágenes que reciba las órdenes del usuario en español, introducidas por medio del teclado. La finalidad de esta práctica es hacer un sistema capaz de interpretar las órdenes introducidas por el usuario y llevarlas a cabo. Dado que la visualización y manipulación de imágenes en la pantalla del ordenador no es el objetivo de esta asignatura, para ello se proporcionará una biblioteca, llamada `ImageMagick`, que dispone de una API en C conveniente.

Lenguaje de órdenes El lenguaje de órdenes se ha diseñado de modo que sea lo bastante complejo como para permitir una cierta variabilidad en las instrucciones indicadas por el usuario. Los siguientes son algunos de los tipos de mandatos que se prevén:

- Mostrar una fotos.
- Avanzar por el álbum, hacia delante o hacia atrás, cualquier número de fotos..
- Ampliar o reducir la foto actual.
- Salir del programa.

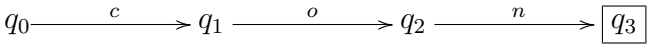
Primera entrega: análisis morfológico En la primera parte de la práctica se pretende la realización de un programa capaz de reconocer independientemente las palabras introducidas por el usuario mediante el uso de Autómatas Finitos Deterministas (AFD). Para ello, definiremos unos pocos lenguajes, según la clase a la que pertenezca cada palabra:

Determinantes	*el, la, los, las, un, una, unos, unas*
Conjunción	*y*
Números	Números enteros (sin ceros a la izda.) Ej: *0, 23, 11, 1943*
Números literales	Números escritos con letras Ej: *uno, cero, cinco, veinticinco*
Adjetivos	*rojo, azul, etc.*
Nombres	*foto, imagen, etc.*
Verbos	*mostrar, enseñar, ver, avanzar, retroceder, seguir, ampliar, reducir, salir*

Además, cada una de las palabras de estos lenguajes pueden tener otros atributos, tales como el número (singular o plural) o el género (masculino o femenino).

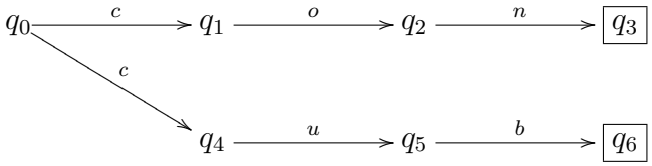
Vamos a considerar que cada clase de palabras es un lenguaje sobre el alfabeto `[a-z0-9]`, y que tenemos cuatro lenguajes más, uno correspondientes a las palabras masculinas, otro correspondientes a las palabras femeninas, otro correspondiente a las palabras en singular, y otro correspondiente a las palabras en plural.

Se trata de hacer un programa que, para cada uno de estos lenguajes, construya automáticamente un AFD que reconozca las palabras que pertenecen a él. Por ejemplo, un autómata que reconoce el lenguaje $\{con\}$ sería el siguiente (con estado inicial q_0):

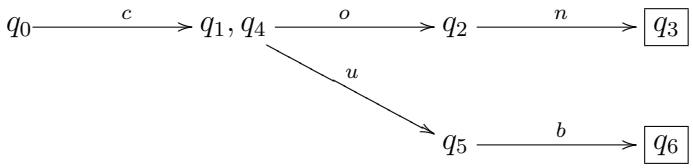


En el caso de que uno de los lenguajes conste de más de una palabra, el procedimiento sería el siguiente:

1. Generar un autómata no determinista para todas las palabras. Por ejemplo, si el lenguaje es $\{con,cub\}$, un posible autómata finito no determinista sería:



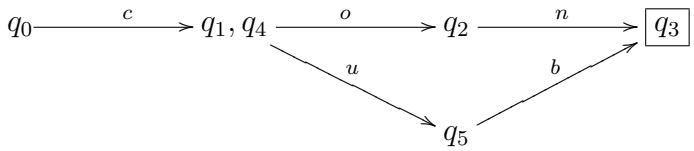
2. A continuación, habría que utilizar el algoritmo para obtener un autómata finito determinista equivalente:



3. **Opcional**: Finalmente, usando el algoritmo de minimización, se obtendría automáticamente un autómata finito determinista mínimo a partir de éste:

el det masc sg
la det fem sg
los det masc pl
las det fem pl
un det masc sg
una det fem pl
unos det masc pl
unas det fem pl
y conj
uno num sg
dos num pl
tres num pl
cuatro num pl
cinco num pl
...
gato nom masc sg
gata nom fem sg
gatos nom masc pl
gatas nom fem pl
...

Figure 1: Ejemplo con el formato del archivo de entrada



Formato del archivo de descripción El programa leerá la descripción de los lenguajes de un archivo. La Figura **??** muestra cómo sería el archivo de entrada. Para cada palabra, se añade el nombre de los lenguajes a los que pertenece. Por ejemplo, la palabra *y* pertenece al lenguaje *conj*, por tratarse de una conjunción; y la palabra *unas* pertenece a los lenguajes *det* (por ser un determinante), *fem* (por ser una palabra femenina) y *pl* (por ser una palabra plural).

Para cada uno de estos lenguajes, el programa debería:

1. Almacenar, para cada lenguaje, todas las palabras que pertenecen a él.
2. Generar un autómata finito no determinista que reconozca ese lenguaje.
3. Transformarlo en un autómata determinista.
4. **Optativo**: Minimizarlo.

5. Informar en *stderr* acerca del número de estados y transiciones que tiene el autómata minimizado.

Para probar el programa, se hará una función principal que, utilizando los autómatas, leerá un texto del teclado y, palabra por palabra, irá mostrando todos los lenguajes a los que pertenece. Por ejemplo, si la cadena de entrada es `mostrar la siguiente imagen`, el resultado será:

```
mostrar verbo
la det fem sg
siguiente adj
imagen nom fem sg
```

Entrega

La entrega electrónica consistirá en un archivo **tar.gz** que incluya:

- El código fuente de los programas.
- Un archivo Makefile para la compilación.
- Un archivo LEEME.TXT con la información solicitada en las normas de prácticas y la hoja de consejos de programación en C, que están disponibles en la página web de prácticas de la asignatura,

  http://www.eps.uam.es/~ealfon/talf

En papel, y **por duplicado**, se entregará la memoria de la práctica, donde se describirán las decisiones de diseño aplicadas, y diagramas de estados de los automátas (si es pertinente).